ARMY RESEARCH LABORATORY

Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs) Implemented Into a Combat Model (A Case Study of the Direct Fire Module II)

Geoffrey C. Sauerborn

ARL-MR-227

June 1995

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

NOTICES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1995	Final, Apr 94 - Mar 95

4. TITLE AND SUBTITLE
Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs) Implemented Into a Combat Model (A Case Study of the Direct Fire Module II)

5. FUNDING NUMBERS
PR: 1L162618A480

6. AUTHOR(S)
Geoffrey C. Sauerborn

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WT-WE
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-227

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

Technology for an individual soldier simulation interacting with a separate combat model simulation was demonstrated in 1995. This memorandum reports how Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs) were used to integrate the soldier simulator and the computer-generated opposition forces into a seamless virtual environment.

14. SUBJECT TERMS

virtual reality, Distributed Interactive Simulation (DIS), Protocol Data Units (PDUs), man in the loop, soldier, Computer Generated Forces (CGF)

15. NUMBER OF PAGES
26

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

The U.S. Army is focusing more research on concept evaluation and simulation as opposed to hardware prototyping and testing. Two simulation tools that can be used for concept evaluation are the Untethered Land Warrior (UTLW) and the Direct Fire Module (DFM) combat simulation. In 1995 these two Army tools were joined into a single simulation. This simulation took place in the form of a demonstration at the Army Research Laboratory (ARL), Aberdeen Proving Ground site. Distributed Interactive Simulation (DIS), a communications protocol specifically designed to merge dissimilar simulation models, was used to join the UTLW and DFM in the 1995 demonstration. This report describes how DIS protocol data units were implemented in the DFM for the demo.

1.1 The Untethered Land Warrior (UTLW)

UTLW is an ARL program sponsored by the Advanced Computation and Information Science Directorate (ACSID). This simulation tool consists of an individual soldier simulator wherein the soldier is presented with a computer-generated image of a virtual battlefield environment. In this environment, the soldier can independently navigate and exercise certain "combat" actions (i.e., fire a weapon at enemy soldiers or vehicles). The enemy may be computer-generated forces or other humans interacting with their own simulators. Whatever their source, all battle participants (clients) are merged into the common simulation via DIS protocol communicated through a computer network.

A milestone for UTLW's first year development was a demonstration of its capability. In this 1995 demonstration, DFM provided the computer-generated opposition forces facing the UTLW soldier.

1.2 The DFM Combat Simulation

The DFM is a portion of the Variable Resolution Combat Model (VRCM) program, which is an ARL program sponsored by the Weapons Technology Directorate. DFM simulates combatants fighting in a direct-fire skirmish (a confrontation where opponents usually have line-of-sight and are close enough to see each other). DFM was designed for two major purposes:

1. To take advantage of the advent of continuous, highly detailed terrain via Variable Resolution Terrain (VRT) (Wald and Patterson 1992). This is accomplished by integrating VRT into the combat model.

2. To be used as a submodel of a larger program simulating a larger battlefield. When the DFM submodel is called from the larger program, it would simulate a smaller clash between direct fire opponents as part of the overall larger battle. The concept of this larger battlefield simulation and its component parts (modules) is collectively called the VRCM (Wald 1994).

1.3 Distributed Interactive Simulation (DIS)

Testing and training with equipment has long proven its worth (especially when equipment is evermore complex and expensive to buy and operate). Joint service and inter-service combined-arms field exercises are even more expensive, but a necessary ingredient to our current defense structure. However, many combined-arms training benefits can be realized, at reduced risk and expense, through simulators interacting with each other on a common virtual battlefield.

DIS refers to a protocol standard specifically designed to facilitate communication between heterogeneous simulators built by different manufacturers. These simulators can be dispersed over a wide geographical region. Using DIS protocol, simulators (which are usually linked via a computer network) run interactively and concurrently on the same virtual battlefield. Linking simulators from many different unit structures with DIS is much less expensive than building a separate, large special purpose combined-arms/joint-service simulator complex, especially since simulators already exist for most major operational equipment. DIS has wide acceptance and is continuing to be refined.

2. THE PHILOSOPHY OF DIS

The philosophy behind a DIS simulation is that all important events are reported to all simulation participants. It is then the responsibility of each participant to use that received information in a manner that is consistent and will enhance the simulation exercise's realism. Important events on the simulated battlefield must be communicated to all DIS participants with 100% truth. It is up to the individual simulators to take that truth and filter it to the perspective of its simulated entities. The Protocol Data Unit (PDU) is the DIS format for truth

conveyance. A PDU is an unambiguous data format for communicating a particular event or specific piece of information.

3. DEMONSTRATION DESCRIPTION

Figure 1 displays the demonstration layout. The Simulation Technology Assessment System (STAS) is a software tool for building combat model generic scenario descriptions. The resulting tactical scenario was used by DFM to generate its simulated forces and their general actions. ARL Stealth and DFM Map are applications for visualizing the battlefield in real time. DFM Map accomplishes this with a two-dimensional (2-D) "map" view of the battlefield where units appear as icons. ARL Stealth displays a three-dimensional (3-D) world as viewed from the stair-stepper's perspective. In ARL Stealth, combatants are displayed as computer-generated images of themselves (e.g., an M1 tank appears as an M1 tank instead of a symbolic icon).[1]

```
+------+         __________          +-------+
|      |        /          \         |       |
|      |       / Tactical   \        |       |
| STAS |---->{               }------>|  DFM  |
|      |       \ Scenario   /        |       |
+------+        \__________/         +-------+
                                       | ^
                                       | |
                                       V |
============================================== DIS Net
        | ^              | ^              | ^
        | |              | |              | |
   +----|--|-------------|--|----+        V |
   |    V  |             V  |    |     +--------+
   |  +--------+     +--------+  |     |        |
   |  | Stair- |     |  ARL   |  |     |  DFM   |
   |  | Stepper|     | Stealth|  |     | 2D Map |
   |  +--------+     +--------+  |     |        |
   |                             |     +--------+
   |            UTLW             |
   +-----------------------------+
```

Figure 1. Physical architecture of the demonstration.

Physically, all of the demo applications and devices were connected on the same local area Ethernet network. Through this link, applications communicated using DIS PDUs. The stair-stepper and DFM passed each other DIS PDUs

[1] Further information on STAS, DFM map, and ARL Stealth can be found in "Simulation for Technology Assessment System (STAS) Life Cycle Demonstration - Phase I" ARL Memorandum Report (to be published).

containing information relating to what their simulated entities were doing and where they were on the virtual battlefield. DFM Map and the ARL Stealth monitored and digested this network traffic. They used this information to visually display the battle as it developed, showing movements, weapons fired, detonations, damaged units, etc. Upon commencing the demo, DFM placed units on the virtual battlefield and began simulating them. The stair-stepper soldier had already been alone on the battlefield, however, now he could interact with the other combatants. The soldier could observe the computer-generated forces through the ARL Stealth display (i.e., if they were within sight) and he was free to take defensive or offensive actions as he saw fit.

Special consideration was made for the individual soldier simulator (the stair-stepper device). At the time of the demo, the stair-stepper had physical controls that allowed the soldier to advance forward, maneuver left or right, and fire his weapon. The weapon fired in the direction he was oriented. Unfortunately, at this point in the UTLW project, there were no physical controls to elevate or depress his weapon's aim. For this reason, the man-in-the-loop would almost certainly never hit a target, except perhaps by accident, since he had no control to superelevate his weapon to the correct target range. Furthermore, software controlling the stair-stepper did not keep track of where its fired munitions flew, and, consequently, could not issue a Detonation PDU when it impacted something.

DFM was modified to overcome these temporary deficiencies in the following manner. When DFM detected a fire event from the stair-stepper (a Fire PDU), it first looked to see if the soldier was facing towards any potential targets. If not, then DFM reissued the Fire PDU with the launch vector field set to the maximum range for the weapon (a LAW II shoulder-launched rocket in this case). If, on the other hand, the soldier was facing any potential targets, DFM then decided which target was in range and at the same time closest to the soldier's orientation (direction he was facing). If a target was so qualified, DFM used that target's range to calculate a fire control superelevation. Then DFM reissued the Fire PDU using this superelevation solution. The launch orientation was still set for the direction the soldier was facing. In this way, the soldier could hit any target in range, provided he was correctly aiming at it. Following the fire event, DFM kept track of where the round flew. Upon the munition striking another entity, or the terrain, DFM issued a Detonation PDU on behalf of the stair-stepper. (It should be noted that this modification to DFM violates DIS protocol. DIS requires the application [the stair-stepper] to control its own munitions and forbids other applications [DFM] from counterfeiting another's identity.)

The Detonation PDU triggered various things to happen in other demo applications. For instance, DFM Map would place a flashing "detonation" icon at the point of impact and draw a line from there back to the stair-stepper icon in

order to indicate that icon was responsible for this detonation. ARL Stealth presented a visual detonation effect. DFM actuated a sound server to simulate the explosion. (Even though DFM sent out the Detonation PDU, the PDU was sent on behalf of the stair-stepper. Therefore DFM behaved as if it had no prior knowledge until it was notified of the detonation event.) DFM also evaluated where the detonation occurred, and if one of its entities was struck, it conducted appropriate vulnerability assessment against that entity. If damage resulted, DFM would issue an Entity State PDU to reflect that damage. (This, in turn, might trigger the ARL Stealth to present further visual effects, depending on the resulting damage level.)

4. PDUs USED IN THE DEMO

Only a small subset of all PDUs in the DIS 2.0 standard were used in the demo. This is because, at the time of the demo, only those aspects of combat were modeled in the DFM or UTLW. Tables 1 and 2 display all the PDUs that the DIS 2.0 IEEE standard contains **[DIS1,DIS2,DIS3]**. These tables are intended to provide a general explanation of each PDU's function. Table 1 lists PDUs intended for entity interaction. Those PDUs incorporated into the demo (Entity State, Fire, and Detonation) are shown in **bold**.

Table 1. DIS 2.0 Entity Interaction PDUs

PDU Name	PDU Function
Other*	* - This PDU is not part of the DIS2.0 standard. It is reserved for experimentation and communicating data not addressed by the standard.
Entity State	The foundation of any DIS exercise, this PDU communicates an entity's identification, location, change in location, orientation, parts movement, damage condition, markings, appearance, etc. Basically the who, what, and where for any entity.

Table 1. DIS 2.0 Entity Interaction PDUs (cont)

PDU Name	PDU Function
Fire	When any weapon is fired, this PDU provides the identification, location, velocity, intended target, and intended range. For burst fire weapons, the number of rounds released in the burst is specified.
Detonation	When a mine explodes, or a munition impacts, the entity controlling it issues a Detonation PDU. This PDU gives the general results of the detonation, but not the resulting damage incurred. It is the impacted entity's own responsibility to assess damaged inflicted upon itself.
Collision	The collision PDU is issued by an entity when it determines it has made physical contact with another entity.
Service Request	Meant for logistical support, this PDU is used to communicate from one entity to another a request to be resupplied or serviced.
Resupply Offer	In response to a service request, the Resupply Offer PDU informs the requesting entity of the amount and type of supplies or service(s) that can be provided.
Resupply Received	This PDU is used for an entity to acknowledge the receipt of supplies or repairs.
Resupply Cancel	An entity receiving supplies issues a resupply cancel to interrupt the resupply in progress.
Repair Complete	The repairing entity serves notice, via this PDU, to the entity being repaired that repairs are complete.
Repair Response	This PDU is used by the entity being repaired to acknowledge the "Repair Complete" PDU.

Table 1. DIS 2.0 Entity Interaction PDUs (cont)

PDU Name	PDU Function
Emission	Electromagnetic warfare, acoustic, and active countermeasure emissions are communicated with this PDU
Laser	Laser operation is reported with the Laser PDU.
Transmitter	The transmitter PDU details information about an electromagnetic spectrum transmitter.
Signal	This PDU can be used to convey data transmitted by a simulated radio signal.
Receiver	This PDU reports the current state of a radio receiver.
IFF	Identify Friend or Foe PDU. This PDU is reserved for future use. Its format is not yet defined.
Expendables	Expendables are items ejected from an entity. The purpose of expendables is usually to confuse a purser. (For example, an airplane may expend chaff to confuse enemy radar; an octopus may expend a cloud of ink to evade a predator). This PDU is reserved for future use. Its format is not yet defined.

5. PDUs NOT USED IN THE DEMO (BUT INCORPORATED IN DFM)

One way to conduct a distributed simulation exercise is to have a single client act as the Simulation Manager (SM). The SM would then be able to exercise great control over the operation and conduct of the simulation. The SM directs other clients when to create entities and where to place them. It can change entities' attributes, reposition them, halt one or all, and even start the entire exercise over. These capabilities have many beneficial and obvious advantages for conducting simulations–especially for instructional purposes. DIS 2.0 provides a set of PDUs through which an SM can be implemented on a

distributed network. Table 2 displays a functional explanation of DIS 2.0's simulation management PDUs. The DFM has partial capability to either act as the SM or comply to the commands of another SM. SM PDUs implemented by the DFM are *italicized* in Table 2. In the 1995 demonstration, however, no client took on the role of SM and, therefore, simulation management PDUs were not used during the demo.

Table 2. DIS 2.0 Simulation Management PDUs

PDU Name	PDU Function
Create Entity	The Simulation Manager (SM) issues this PDU to instruct a client to create an entity. The entity is then placed in a "stopped state" awaiting further instruction.
Remove Entity	SM issues this PDU to instruct a client to remove the entity from the simulation. This changes the entity from a "simulating state" to a "stopped state."
Start / Resume	SM issues this PDU to an entity to instruct it to commence or resume simulating. The entity goes from a "stopped state" to a "simulating state."
Stop / Freeze	SM issues this PDU to an entity to halt its simulating process. The entity goes from a "simulating state" to a "stopped state."
Acknowledge	The Acknowledge PDU is sent from the entity to the SM in response to the Create, Remove, Start/Resume, and Stop/Freeze PDUs.
Data Query	SM issues this PDU to find out details concerning the internal variables or state of an entity.

Table 2. DIS 2.0 Simulation Management PDUs (cont)

PDU Name	PDU Function
Set Data	SM issues this PDU to change an entity's internal variables or state.
Data	This PDU is issued by the entity to an SM in response to the Data Query or Set Data PDU. If in response to the Data Query, the entity reports its internal values for the variables queried by the SM. If in response to the Set Data PDU, this PDU contains the new values after the entity has finished changing them.
Action Request	SM issues this PDU to request an entity to preform a particular action. Some examples of actions that can be requested: locally store specified entity internal information, report to the SM when a certain damage level is reached, and report to the SM when out of ammo.
Action Response	An entity acknowledges the SM's Action Request PDU with an Action Response PDU.
Event Report	When an SM-requested action occurs (e.g., "ran out of ammo"), the entity informs the SM with an Event Report PDU.
Message	This PDU can be used to pass arbitrary messages (text strings) to any entity, or the SM.

6. HOW DEMO PDUs ARE IMPLEMENTED

Inasmuch as is our understanding of the DIS 2.0 standard (22 March 1993 second draft), all PDUs were used in a manner which was in compliance with DIS. We now explain in greater detail how each PDU used in the demo is implemented. Only those PDUs that were used in the demo (the Entity State, Fire, and Denotation PDUs) are addressed. For a general itemization and explanation of the kinds of data contained within these PDUs, refer to the appendix.

6.1 Entity State PDU

The Entity State PDU is the backbone of any DIS exercise. Its purpose is to give almost all important information about an entity. This PDU identifies whether the entity is an enemy, friendly, or neutral. The entity type (e.g., tank) and its specific model are also contained here. The entity's location, orientation, velocity, and acceleration are in the Entity State PDU. Its appearance is denoted here (e.g., dust cloud kicked-up behind a moving vehicle entity, or perhaps the entity is on fire and burning). An entity tells the DIS world its damage state via this PDU. Moving parts are denoted, as is their current position and motion. Basically just about everything other DIS clients need to know about an entity can be found in the Entity State PDU.

DIS 2.0 specifies a technique known as "heartbeat" to issue PDUs. A "heartbeat" is when all entities participating in the simulation transmit an Entity State PDU on a regular interval. The interval is agreed to beforehand by simulation participants (once every couple of seconds was used for the demo). When an entity has not been heard from (via an Entity State PDU) for a certain amount of time (again, that time being agreed to beforehand), clients are to consider that entity no longer in existence. Using a heartbeat with a short time interval is not necessary and will load the DIS network with much more data than required, but it is a useful technique when first implementing a DIS simulation since other clients will quickly know if one client stops functioning. Also, in a larger DIS exercise with many entities and many clients, the exercise can continue smoothly even if clients exit from the battle, since their simulated entities will not be orphaned, but rather exit with them. This will avoid many anomalies in the exercise, such as the case where entities expend large amounts of ammunition against orphaned entities, which incidentally have become impervious to all munitions. (We will see why orphaned entities are invincible when we address the Detonation PDU.)

DIS implements a method of tracking entities know as "Dead Reckoning." In Dead Reckoning, other clients track the location of an entity based on the last emitted Entity State PDU. Between emissions of PDUs, clients can predict the entity's location based on the last known location, velocity, and acceleration. Using this same data, the entity itself also keeps track of what other clients perceive its current location to be. When the difference between an entity's actual location and its Dead Reckoned location become too great, the entity emits a fresh Entity State PDU. The Dead Reckoning error threshold for emitting a new PDU is agreed to beforehand by simulation participants.

6.1.1 What is Modeled as an Entity

It is up to the DIS client to decide what it wishes to simulate as an entity. In addition to the normal entities one might expect (such as a tank, a soldier, a truck), there are other less obvious entities—a building, a bridge, a tree. In fact, almost anything that can be uniquely identified via the DIS enumeration standard [**DIS2**] can be treated as an entity in the simulation. It is up to the client's discretion.

A terminally guided munition, such as a guided antitank missile, is commonly treated as an entity since its course is frequently changing and it has a long time of flight (relative to direct fire ballistic munitions). Ballistic munitions are not usually treated as entities. This is because they have a very short time of flight and their impact point can be predicted right away (given enough information about launch conditions and flight dynamics).

6.2 Fire PDU

When an entity fires a weapon, a Fire PDU is emitted. In the same way that the Entity State PDU conveys almost all that is needed to be known concerning an entity, the Fire PDU tells most of the data needed to be known about a fired munition.

Under DIS 2.0, it is the responsibility of the firing entity to keep track of where its munition goes and, hence, what happens to it. If and when the munition impacts something (whether the ground or another entity), the firing entity emits a Detonation PDU. It is then the responsibility of any entities who are affected by the detonation to respond in a correct manner.

6.3 Detonation PDU

An impacting round or exploding bomb, mine, or munition is announced by a Detonation PDU. The Detonation PDU represents the end of a munition's path and its existence. When notice of a detonation is received, entities determine if they are affected by the event and respond accordingly. If the detonation influences an entity, it is the responsibility of that entity to change its internal state in a way that reflects the result of the detonation. It is also the affected entity's responsibility to inform all clients of the result, via an Entity State PDU. This is why orphaned entities become indestructible; their parent simulator (or

client) has left the simulation and cannot respond properly to the Detonation PDU.

7. SUMMARY

Portions of the Distributed Interactive Simulation (DIS) Version 2.0 IEEE Draft Standard were used as a means to communicate important battlefield events and environmental information for a 1995 demonstration of the Untethered Land Warrior (UTLW) technology. In this demonstration, a human (man-in-the-loop) interfaced with the UTLW simulator. The simulator presented a "virtual reality" battlefield environment for the soldier. The human could freely maneuver and take actions within this environment. Within this same virtual world were opposition forces. These computer-generated enemy forces were controlled by the Direct Fire Module (DFM).

DIS Protocol Data Units (PDUs) were used in the DFM and the UTLW to link events that occurred into a seamless battle simulation to the extent that neither the man-in-the-loop nor the computer-generated forces knew that the other was a separate and self-contained system. Rather, the two systems merged and become one battle simulation within this "virtual reality" environment.

8. CONCLUSIONS

The UTLW and DFM have demonstrated their capability to interact with DIS compliant applications. This alone demonstrates their potential to be used in concept evaluation by interacting with or "plugging into" a growing pool of DIS compliant models. The environment and capabilities of the DFM and UTLW should continue to be expanded, developed, and refined.

DIS is an effective, efficient, and economical way to join dissimilar simulation tools. It is flexible enough to incorporate experimental weapons and tactical concepts into simulations. Being an IEEE draft standard, it has wide acceptance and is under continuing revision and therefore has great potential to keep up with newly discovered requirements. ARL should continue to expand its capability in DIS. Doing so can enhance the attractiveness of ARL products that have DIS capability already built-in.

REFERENCES

DIS1[2], "Proposed IEEE Standard Draft: Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications", Institute for Simulation and Training, (DIS Version 2.0 second draft, 22 March 1993) IST-CR-93-01

DIS2[2], "Enumeration and Bit Encoded Values for use with Protocols for Distributed Interactive Simulation Applications", Institute for Simulation and Training, (DIS Version 2.0 second draft, 22 March 1993) IST-CR-93-02

DIS3[2], "Rational:Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications", Institute for Simulation and Training, (DIS Version 2.0 second draft, 22 March 1993) IST-CR-93-03

Wald J.K., Patterson C.J. "A Variable Resolution Terrain Model for Combat Simulation," U.S. Army Ballistic Research Laboratory, APG, MD, BRL-TR-3374. July 1992

Wald J.K. "Time History Injection in a Nested System of Battlefield Simulation Models," U.S. Army Ballistic Research Laboratory, APG, MD, BRL-TR-2984. March 1994.

2 - (Collectively DIS1,DIS2, and DIS3 make up the IEEE draft standard known as "DIS version 2.0" or "DIS 2.0")

INTENTIONALLY LEFT BLANK.

ACRONYMS

ACSID	Advanced Computational Science and Information Directorate
ARL	Army Research Laboratory
DIS	Distributed Interactive Simulation
DFM	Direct Fire Module
IEEE	Institute of Electrical and Electronic Engineers
PDU	Protocol Data Unit
UTLW	Untethered Land Warrior
WTD	Weapons Technology Directorate

INTENTIONALLY LEFT BLANK.

APPENDIX

EXPLANATORY USE OF DIS PDUs EMPLOYED IN THE DEMO

INTENTIONALLY LEFT BLANK.

The PDUs used in the demo are tabulated in this appendix. Fields are grouped by their subject content. Subfields, actual binary format, and size of each field are not presented here. (See references **DIS1,DIS2,DIS3** for these kind of details). The purpose here is to present the kind of data PDUs contain in an explanatory format that will benefit anyone—not just the systems programmer. In Tables A1, A2, and A3, the Entity State, Fire, and Detonation PDUs are divided into subject matter content, respectively. A terse explanation is given for each subject.

Table A1. Entity State PDU Content

Subject Area		**Explanation**
1	Protocol Header	This area of the PDU identifies the PDU type and other application administrative data required to process this PDU. Every PDU type has a Protocol Header.
2	Entity ID	This area of the Entity State PDU uniquely identifies the entity sending this PDU.
3	Force ID	Force ID identifies the entity as being a "friend," "foe," "neutral," or "other." This is the absolute truth concerning whose "side" the entity is on and is not necessarily the *perception* others will have concerning this entity's identity. (This allows room for mistaken identification: see Entity Type and Alternate Entity Type).

Table A1. Entity State PDU Content (cont)

Subject Area		Explanation
4	Entity Type	This describes how comembers of the entity's own force will "see" this entity. This field specifies exactly what the entity is. There is a wide spectrum of entities that can be specified using the DIS 2.0 standard. (Some examples are M1A2 Abrams main battle tank, M47 Dragon anti-tank missile, Iowa Class Battleship BB62 USS New Jersey, a school of shrimp, a dismounted Swiss infantry soldier with a Soviet 7.62-mm SVD Sniper Rifle.)
5	Alternate Entity Type	This describes how other members not on the same force as the entity will "see" this entity. Normally this field specifies exactly what the entity is.
6	Location	Where the entity is located.
7	Velocity	Entity's linear velocity.
8	Orientation	In what direction the entity is oriented (facing).
9	Dead Reckoning Data	Provides the data necessary for other entities to accurately track the entity using "dead reckoning." It not only provides data to track the entity's position, but also changes in the entity's orientation and movement of parts attached to the entity.

Table A1. Entity State PDU Content (cont)

Subject Area		Explanation
10	Appearance and markings	This area of the PDU is used to add details concerning the entity's appearance. Types of details that can be described are paint scheme (camouflage, or other), appearance of specific types of damage, issuance of trailing effects (dust cloud, rocket plume, contrails, etc.), running lights, flash of guided munition being launched, posture (kneeling, prone, upright, destroyed), unique markings (i.e., bumper number, country symbol), and others.
11	Capabilities	Specifies certain capabilities this entity is able to perform.
12	Articulation Parameters	This area specifies in detail any parts that are attached to the entity. Included are both those parts that can be removed (such as an air to air missile attached under a jet's wing) and those parts that cannot be removed but can be articulated (such as the periscope of a submarine or a tank's turret). Specified are all data concerning which parts are attached to the entity (or to each other) how they move, their current position, orientation, change in position and, rate of change.

Table A2. Fire PDU Content

Subject Area		Explanation
1	Protocol Header	This area of the PDU identifies the PDU type and other application administrative data required to process this PDU. Every PDU type has a Protocol Header field.
2	Firing Entity	Uniquely identifies who is firing this munition.
3	Target	Identifies the intended target (if any).
4	Munition ID	This area uniquely identifies the munition fired (if the munition is to be tracked).
5	Event ID	The Event ID uniquely identifies this particular fire event and will be used again in the simulation when other events directly related to this fire event occur (e.g., the Detonation, and Event Record PDU).
6	Location	Exactly where the munition emanates from (the point of origin).
7	Velocity	Initial velocity vector.
8	Burst Descriptor	This area specifies the number of rounds fired, the rate of fire, and the type of munition, warhead, and fuse.
9	Range	This is the range that the firing entity has assumed in computing its fire control solution.

Table A3. Detonation PDU Content

Subject Area		Explanation
1	Protocol Header	This area of the PDU identifies the PDU type and other application administrative data required to process this PDU. Every PDU type has a Protocol Header.
2	Firing Entity	Uniquely identifies who fired this munition.
3	Target	Identifies the targeted entity (if any).
4	Munition ID	This area uniquely identifies the munition fired (if the munition is to be tracked).
5	Event ID	The Event ID uniquely identifies other events that are related to this one. (For example, if the detonation is the result of a weapon being fired, the fire PDU's "Event ID" will have been the same value found in this field).
6	Location	Exactly where the detonation occurs (the point of impact).
7	Velocity	Velocity vector of munition at the time of detonation.

Table A3. Detonation PDU Content (cont)

	Subject Area	Explanation
8	Burst Descriptor	This area specifies the number of rounds fired, the rate of fire, and the type of munition, warhead, and fuse.
9	Result	The result tells a little about where the detonation occurs. It can be used to denote direct or proximate impact on the targeted entity. Or a direct or proximate impact on the ground. It can also be used to communicate the failure of the round or fuse to function and other detonation results.
10	Articulation Parameters	This area specifies the current state and position of any parts on the impacted entity that may be effected by the detonation.

NO. OF COPIES	ORGANIZATION
2	ADMINISTRATOR ATTN DTIC DDA DEFENSE TECHNICAL INFO CTR CAMERON STATION ALEXANDRIA VA 22304-6145
1	DIRECTOR ATTN AMSRL OP SD TA US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR ATTN AMSRL OP SD TL US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR ATTN AMSRL OP SD TP US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
5	DIR USARL ATTN AMSRL OP AP L (305)

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
8	DIR, USARL ATTN: AMSRL-WT-W, C MURPHY AMSRL-WT-WE, G SAUERBORN (5 CP) AMSRL-SC-SS, KENNETH SMITH AMSRL-IS, PHILIP EMMERMAN

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number ARL-MR-227 Date of Report June 1995

2. Date Report Received ______

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ______

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ______

CURRENT ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
U.S. ARMY RESEARCH LABORATORY
ATTN: AMSRL-WT-WE
ABERDEEN PROVING GROUND, MD 21005-5066